

Tracking Control of Redundant Mobile Manipulator: An RNN based Metaheuristic Approach

Abstract

In this paper, we propose a topology of Recurrent Neural Network (RNN) based on a metaheuristic optimization algorithm for the tracking control of mobile-manipulator while enforcing nonholonomic constraints. Traditional approaches for tracking control of mobile robots usually require the computation of Jacobian-inverse or linearization of its mathematical model. The proposed algorithm uses a nature-inspired optimization approach to directly solve the nonlinear optimization problem without any further transformation. First, we formulate the tracking control as a constrained optimization problem. The optimization problem is formulated on position-level to avoid the computationally expensive Jacobian-inversion. The nonholonomic limitation is ensured by adding equality constraints to the formulated optimization problem. We then present the Beetle Antennae Olfactory Recurrent Neural Network (BAORNN) algorithm to solve the optimization problem efficiently using very few mathematical operations. We present a theoretical analysis of the proposed algorithm and show that its computational cost is linear with respect to the degree of freedoms (DOFs), i.e., $O(m)$. Additionally, we also prove its stability and convergence. Extensive simulation results are prepared using a simulated model

Email addresses: `ameer.h.khan@connect.polyu.hk` (Ameer Hamza Khan),
`shuaili@ieee.org` (Shuai Li), `chdchao@sina.com` (Dechao Chen), `liaolf@126.com` (Liefu Liao)

of IIWA14, a 7-DOF industrial-manipulator, mounted on a differentially driven cart. Comparison results with particle swarm optimization (PSO) algorithm are also presented to prove the accuracy and numerical efficiency of the proposed controller. The results demonstrate that the proposed algorithm is several times (around 75 in the worst case) faster in execution as compared to PSO, and suitable for real-time implementation. The tracking results for three different trajectories; circular, rectangular, and rhodonea paths are presented.

Keywords: Mobile-manipulator, Tracking Control, Metaheuristic Optimization, Recurrent Neural Network, Nature-inspired Algorithm, Redundancy Resolution

1. Introduction

Mobile robotic manipulators have found interesting applications in industry as well as in rescue and military operations [1, 2] because of their mobility, agility, and flexibility. Modern innovations in robotics and control theory combined with the increasing power of computing systems have driven the focus of academic research toward mobile robots [3, 4]. Robotic manipulators are used in industry to perform a several type of tasks e.g. moving, assembling, and transporting objects [5, 6, 7]. Most of industrial manipulators have redundant joints [8, 9] because the extra degree-of-freedom (DOF) allows to meet additional design objectives [10, 11] e.g. obstacle avoidance [12], joint-angle limits and torque minimization [13]. Mounting the manipulator on a mobile platform is more desirable because it further increases the DOF of the robotic-platform. However, for such a redundant mobile-manipulator, inverse-kinematic modeling and tracking control becomes challenging. Firstly, for any task-space trajectory, an infinite number of trajectories exist in joint-space. Secondly, the transformation from task-space to joint-space cannot be expressed in closed-form [14, 15].

For redundant nonmobile manipulators (i.e., manipulators with fixed base), the problems of tracking control is well-studied in the control literature [16, 17, 18, 19, 20, 21]. One of the traditional method for resolving the joint-space re-

dundancy (i.e. calculating an optimal joint-space trajectory) is called Jacobian-matrix-pseudo-inverse (JMPI) [22, 23, 24]. However as shown by Klein *et al.* [25] JMPI does not demonstrate predictable behaviour and can lead to undesirable joint-space configurations. Additionally it requires the calculation of pseudo-inverse of Jacobian matrix at each time-step, which is a computationally intensive process. Moreover, it cannot be used in the presence of inequality constraints. Modern control-algorithms use optimization-driven frameworks to solve the redundancy resolution problem [26, 10, 11, 27]. The optimization-driven approaches allow to incorporate an arbitrary design objective into the tracking control framework by properly formulating a constrained optimization problem. Woolfrey *et al.* [28] proposed an algorithm to minimize the torque of the industrial manipulators. Neural network based manipulator controllers are proposed by Wei *et al.* [29] and Wang *et al.* [30], which work by numerically solving the optimization problem in real-time. Similarly, Li *et al.* [16] proposed an optimization-based dual Recurrent Neural Network (RNN) algorithm for controlling multiple manipulators simultaneously. Control algorithms based on Jacobian matrix estimation [31, 32] and adaptive control are also proposed [33, 34, 35, 36, 37, 38, 39].

For the tracking control of redundant mobile-manipulators (also discussed in this paper), many approaches have also been proposed during recent years [40, 41, 42]. For example, Xiao *et al.* [41] proposed a neuro-dynamical approach for robust stabilization of a mobile-manipulator. Miah *et al.* [43] proposed a kinematic controller for a nonholonomic mobile platform considering model uncertainties and sensing errors. Connell *et al.* [40, 42] employed Rapidly exploring Random Tree (RRT) algorithm for dynamic path planning of mobile robots. Chwa [44] proposed fuzzy-adaptive controller for the tracking control of mobile robots under slippage and unknown external disturbances. Deepak *et al.* [45] proposed a controller for wheeled mobile manipulator using an artificial immune system for transporting objects and tools in an industrial environment. Dechao and Yuong [46] have proposed an adaptive zeroing neural network for the kinematic control of redundant manipulator mounted on a differentially

driven cart. However, most of these algorithms design the controller on velocity level, which also requires computationally expensive operation of Jacobian pseudo-inverse.

In this paper, we propose a novel RNN architecture inspired by a metaheuristic optimization algorithm; called Beetle Antennae Olfactory Recurrent Neural Network (BAORNN), for the tracking control of the mobile-manipulators. We first formulate the tracking control as a constrained optimization problem. Apart from the tracking control, the formulated tracking controller also incorporates the joint-limit constraint. We then propose the BAORNN algorithm to solve the formulated optimization problem in real-time. In the formulation of the BAORNN algorithm, we take advantage of well-known characteristics of metaheuristic optimization algorithms [47, 48, 49], i.e., their ability to numerically solve complex, nonlinear and nonconvex optimization problems in a computationally efficient manner [50, 51, 52]. The metaheuristic optimization has not only been a focus of theoretical interest, but they have also found several applications [53, 54, 55, 56] in real-world scenarios in recent years. The original BAO algorithm, as proposed by Jiang and Li [57, 58], is a nature-inspired metaheuristic optimizer, inspired by the food foraging behavior of beetles. Although recently proposed, BAO has found potential applications in several real-world scenarios [59, 60]. A peculiar feature of the BAORNN algorithm is the concept of “virtual robots”, a feature unique to the proposed algorithm as compared to traditional algorithms, which allows anticipating the consequences of the joint control-actions. By using virtual robots, the algorithm ensures that the real manipulator only moves when they satisfy the constraints of the optimization problem. Additionally, formulation of the algorithm as RNN will enable fast prototyping by leveraging the hardware parallelism, distributed processing, and software optimizations, offered by modern computing frameworks.

It must be noted that the concept of “virtual robots” allows the formulation of the kinematic controller at the position-level, in contrast to the velocity-level control as done in other works. The remove the necessity of calculating the pseudo-inverse of the Jacobian matrix and increase the computational efficiency

of the controller. For generating the results, we used the computation model of IIWA14 industrial manipulator. The manipulator is mounted on a differentially driven mobile platform. We performed the simulations under different values of hyper-parameters and analyzed their effect on the tracking performance of the mobile-manipulator. We also presented the comparison results of the proposed algorithm using Particle Swarm Optimization (PSO) [61] algorithm as a benchmark. PSO is a popular metaheuristic optimization algorithm, which has been widely studied and tested in several practical scenarios [62, 63]. Therefore, PSO is an algorithm of choice when comparing the performance of a newly formulated optimization algorithm. The main highlights of this paper are listed as follows

1. A kinematic controller is formulated for tracking the time-varying reference trajectory. Unlike most traditional algorithms, the proposed controller is formulated at the position-level to avoid the Jacobian matrix pseudo-inversion and enhance computational efficiency.
2. The tracking control of a generic mobile-manipulator is considered and it is formulated as a constrained optimization problem.
3. A metaheuristic based RNN is proposed to solve the constrained optimization problem numerically.
4. Theoretical analysis is presented to prove the stability and convergence of the BAORNN algorithms.
5. We performed extensive numerical analysis on IIWA14 (KUKA LBR) to verify the efficacy of the proposed algorithm.

The remainder of this paper is organized as follows. Section 2 describe the tracking control problem for redundant mobile-manipulators along with formulation joint-angle limits. In Section 3, the formulation and theoretical analysis are presented. Section 4 outlines the simulation methodology and present the results along with discussion. Section 5 concludes the paper.

2. Problem Formulation

In this section, the tracking control problem for redundant mobile-manipulator is mathematically formulated as a constrained quadratic optimization problem along with the joint-angle constraints. Additionally, we will develop the model for our differentially driven mobile-platform.

2.1. Tracking Control

Consider an industrial mobile robot placed on a differentially driven platform as shown in Fig. 1. If the mobile-manipulator is required to move an object along a specified trajectory in cartesian task-space, i.e., the position of end-effector must follow a reference trajectory; then the tracking control deals with the calculation of a corresponding trajectory in joint-space to follow the task-space trajectory. Suppose the manipulator have m total joints. Since the manipulator is mounted on a differentially driven cart, which contribute three more parameters, \bar{x} , \bar{y} , and $\bar{\psi}$, in determining the position of the end-effector. The mapping from joint-space to task-space is defined as

$$\mathbf{x} = F(\Theta), \quad (1)$$

where $F(\cdot)$ is the forward-kinematic mapping of the mobile-manipulator, $\Theta \in \mathbb{R}^{m+3}$ is the vector of joint-space coordinates and $\mathbf{x} \in \mathbb{R}^3$ is the position of end-effector in task-space coordinates. $m > n$ for a redundant manipulator. Note that the term “joint-space coordinates” is used to refer the combination of joint-angles and the current state of differentially driven robot. The joint-space coordinate vector is defined as $\Theta = [\bar{x} \ \bar{y} \ \bar{\psi} \ \boldsymbol{\theta}]^T$. Where $\boldsymbol{\theta} \in \mathbb{R}^m$ is the vector of joint-angles. Based on the structure of the mobile platform in Fig. 1, the forward-kinematic model in (1) can be expressed as,

$$\mathbf{x} = [\bar{x} \ \bar{y} \ L]^T + \mathcal{R}_z(\bar{\psi})f(\boldsymbol{\theta}) \quad (2)$$

where $f(\cdot)$ is the forward-kinematic model of the manipulator without the mobile base. $\mathcal{R}_z(\cdot)$ is the rotation matrix around the z-axis by an angle of $\bar{\psi}$. L is a

constant factor denoting the height of the mobile platform, as illustrated in Fig. 1. The forward-kinematic models $F(\cdot)$ and $f(\cdot)$ are nonlinear and depends on the Denavit–Hartenberg (DH) parameters of the robotic-manipulator and the mechanical parameter of the mobile platform. Although the mapping function $F(\cdot)$ is trivial to calculate for a serially linked robot. However, the reference trajectory for end-effector is usually specified in the cartesian task-space coordinates; therefore, we are more interested in the inverse-kinematic mapping. An inverse-kinematic model can be defined based on 1

$$\Theta = F^{-1}(\mathbf{x}), \quad (3)$$

where $F^{-1}(\cdot)$ is the inverse function. Lets denote the time-varying reference trajectory as $\mathbf{x}_r(t)$. To track this trajectory, the resultant joint-space trajectory $\Theta_r(t)$ should satisfy the following relation,

$$\mathbf{x}_r(t) = F(\Theta_r(t)). \quad (4)$$

If the inverse-kinematic model $F^{-1}(\cdot)$ can be expressed in closed-form, then the value of $\Theta_r(t)$ can be expressed analytically. However, due to the nonlinear nature of $F(\cdot)$, such a closed-form expression does not exist; therefore, numerical methods are used to calculate an approximate solution.

In fact, for a redundant mobile-manipulator, given a particular reference trajectory $\mathbf{x}_r(t)$, an infinite number of solutions exist in joint-space $\Theta_r(t)$. We formulate the following objective function to resolve the redundancy

$$\min_{\Theta(t)} \|\mathbf{x}_r(t) - F(\Theta(t))\|_2, \quad (5)$$

to simplify the notation, we define the objective function $g(\cdot)$ as follow,

$$g(\mathbf{x}_r, \Theta) = \|\mathbf{x}_r - F(\Theta)\|_2. \quad (6)$$

Remark 1. *The objective function formulated in (6) only considers the kinematic model of the mobile-manipulator. The controller based on the kinematic model of the manipulator has been extensively studied in several recent works*

[64, 65, 66, 8]. Kinematic controllers are also being used in industrial manipulators such as UR-10 manipulator [67], Quattro-650HS [68], IRB-360 [69], and DOBOT [70].

2.2. Mobile Platform

Now we will mathematically formulate the motion of differential driven mobile platform on which the manipulator is mounted. The structure of the mobile platform is shown in Fig. 1. It can be seen that the platform is cylindrical with a radius of R . The radius of the wheels is denoted by r . The platform is driven by two motors controlling the angular speed of each wheel. $\dot{\phi}_L$ and $\dot{\phi}_R$ represent the angular speed of the left and right wheel, respectively.

According to the kinematics of the differentially driven robots presented in [71], the state variables \bar{x} , \bar{y} , and $\bar{\psi}$ of the mobile platform relates to the control inputs as follow

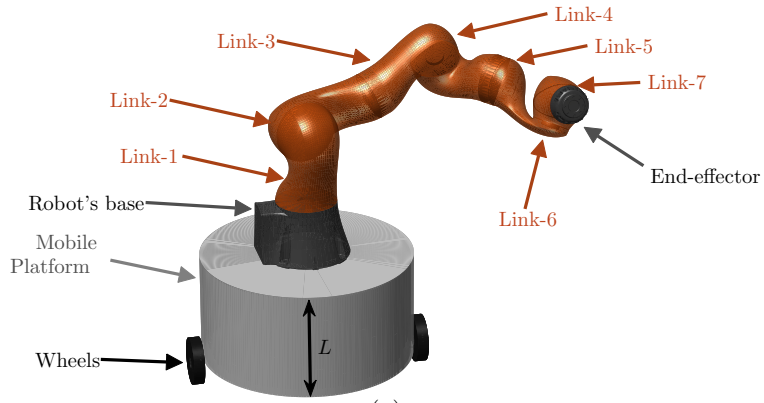
$$\begin{bmatrix} \dot{\bar{x}} \\ \dot{\bar{y}} \\ \dot{\bar{\psi}} \end{bmatrix} = \begin{bmatrix} \frac{r}{2} \cos(\bar{\psi}) + \frac{rd}{2R} \sin(\bar{\psi}) & \frac{r}{2} \cos(\bar{\psi}) - \frac{rd}{2R} \sin(\bar{\psi}) \\ \frac{r}{2} \sin(\bar{\psi}) - \frac{rd}{2R} \cos(\bar{\psi}) & \frac{r}{2} \sin(\bar{\psi}) + \frac{rd}{2R} \cos(\bar{\psi}) \\ -\frac{r}{2R} & \frac{r}{2R} \end{bmatrix} \begin{bmatrix} \dot{\phi}_L \\ \dot{\phi}_R \end{bmatrix}. \quad (7)$$

It must be noted that because of nonholonomic constraint on the differential driven platform, i.e., no slippage of wheels, it is not possible to control the three state variables \bar{x} , \bar{y} , and $\bar{\psi}$ simultaneously using two control inputs $\dot{\phi}_L$ and $\dot{\phi}_R$. Therefore, although the vector Θ is $(m+3)$ -dimensional, the actual DOF of the mobile-manipulator is $m+2$. To resolve this ambiguity, we incorporate the following nonholonomic constraint into our optimization problem, by following the approach of [72],

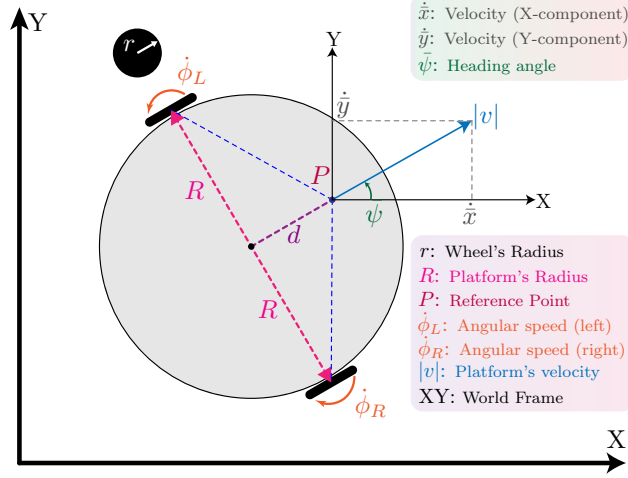
$$(\dot{\bar{x}} + d \sin(\bar{\psi}) \dot{\bar{\psi}}) \sin(\bar{\psi}) - (\dot{\bar{y}} - d \cos(\bar{\psi}) \dot{\bar{\psi}}) \cos(\bar{\psi}) = 0. \quad (8)$$

This constraint reduces the DOF of the forward-kinematic model (1) by one. To simplify the notation, let us define a equality constraint function $h(\cdot) = 0$

$$h(\dot{\bar{x}}, \dot{\bar{y}}, \dot{\bar{\psi}}) = (\dot{\bar{x}} + d \sin(\bar{\psi}) \dot{\bar{\psi}}) \sin(\bar{\psi}) - (\dot{\bar{y}} - d \cos(\bar{\psi}) \dot{\bar{\psi}}) \cos(\bar{\psi}).$$



(a)



(b)

Figure 1: Illustration of a mobile-manipulator. (a) Shows the links of 7-DOF IIWA14 manipulator used to compile the results. (b) Visualize the terms presented in Section 2.2.

The equations (7) and (8) can be simplified if the point P , as shown in Fig. 1, is chosen to lie in the middle of the line joining two wheels, i.e., $d = 0$.

In addition to nonholonomic constraints, a limit on maximum rotational speed is imposed by the mechanical power rating of the motors. These constraints can be formulated as

$$\dot{\phi}_L < \dot{\phi}_{\max} \quad \text{and} \quad \dot{\phi}_R < \dot{\phi}_{\max}, \quad (9)$$

where $\dot{\phi}_{\max}$ denotes the maximum rotational speed of a wheel.

2.3. Joint-Angle Limits

The joints of the motors are designed to work within mechanically optimal limits. The limit depends on the mechanical design of the manipulator and the type of actuator used. Extending a joint beyond its mechanical limits can cause severe damage to the robot as well as the surrounding environment. Here we formulate the joint-angle limits as following inequality constraints

$$\boldsymbol{\theta}^- < \boldsymbol{\theta}(t) < \boldsymbol{\theta}^+, \quad (10)$$

where $\boldsymbol{\theta}^- \in \mathbb{R}^m$ and $\boldsymbol{\theta}^+ \in \mathbb{R}^m$ are the lower and upper limits on the motion of each joint of the manipulator.

2.4. Unified Optimization Problem

In previous subsections, we formulated: tracking control objective function as given in (6), nonholonomic constraints given in (8), rotational speed limit constraints given in (9) and the joint-angle constraints given in (10). Now we combine them into the following constrained optimization problem

$$\begin{aligned} \min_{\boldsymbol{\Theta}(t)} \quad & g(\mathbf{x}_r(t), \boldsymbol{\Theta}(t)) \\ \text{s.t.} \quad & \boldsymbol{\theta}^- < \boldsymbol{\theta}(t) < \boldsymbol{\theta}^+ \\ & \dot{\phi}_L < \dot{\phi}_{\max} \\ & \dot{\phi}_R < \dot{\phi}_{\max} \\ & h(\dot{x}, \dot{y}, \dot{\psi}) = 0. \end{aligned} \quad (11)$$

Remember that the objective function $g(\cdot)$ is same as given in (6). The numerical solution to this problem will provide an optimal joint-space trajectory $\Theta_r(t)$. In the next section, we will formulate the BAORNN algorithm to solve this optimization problem.

3. Control Design

In this section, we present the formulation of BAORNN algorithm and describe its RNN architecture. Then we offer the theoretical analysis of the formulated algorithm.

3.1. BAORNN Algorithm

BAORNN algorithm is a metaheuristic optimization algorithm inspired by the food foraging behavior of beetle. The BAO algorithm [57] inspires its mathematical formulation. Among nature-inspired metaheuristic optimization algorithms, BAO offers a unique advantage that it only uses a single search particle instead of a swarm of particles to solve an optimization problem. Before mathematically formulating the algorithm, let us describe key strategies used by the beetle to search for food in a previously unknown environment. A beetle uses its antennae pair and sense of smell to efficiently find its way toward food source i.e., region of maximum smell. Therefore the task of searching for food is essentially an optimization problem. Instead of moving in random directions, beetle compares the magnitude of smell intensity at both antennae location and uses their difference to guess an optimal direction for taking the next step. The use of the sense of smell to calculate an optimal direction before taking an actual step inspired us to introduce the concept of “virtual robots” (analogous to the antennae’s sense of smell) into the BAORNN algorithm.

For mathematical formulation, consider at time-step k the joint-space coordinates of mobile-manipulator are Θ_k . To model the beetle antennae, generate a normally distributed random direction vector $\vec{\mathbf{b}} \in \mathbb{R}^{m+3}$, such that $|\vec{\mathbf{b}}|_2 = 1$. Since antennae pair is attached on the opposite side beetle, their location can

be written as

$$\Theta_{kL} = \Theta_k + \lambda_k \vec{\mathbf{b}} \quad \text{and} \quad \Theta_{kR} = \Theta_k - \lambda_k \vec{\mathbf{b}}, \quad (12)$$

where Θ_{kL} and Θ_{kR} are the antennae location at time-step k , λ_k is a hyperparameter of our algorithm and denotes the antenna-length. To ensure that these vectors satisfy the constraint in the optimization problem (11) we define a projection function

$${}^\Omega \Theta_{kX} = \mathcal{P}_\Omega(\Theta_{kX}), \quad (13)$$

where $\mathcal{P}_\Omega(\cdot)$ is a function which project the input vector on the constrained set Ω , ${}^\Omega \Theta_{kX}$ is the projected vectors, here $X \in \{L, R\}$. Note that the left superscript symbol is used to denote a projected vector in this paper. The constrained set is defined as

$$\Omega = \{\Theta \in \mathbb{R}^m | \theta^- < \theta < \theta^+ \wedge \dot{\phi}_L < \dot{\phi}_{\max} \wedge \dot{\phi}_R < \dot{\phi}_{\max} \wedge h(\dot{x}, \dot{y}, \dot{\psi}) = 0\},$$

where the terms used in the above expression are the same as defined in Section 2. Next, the value of objective function $g(\cdot)$ is calculated at the projected antennae locations using “virtual mobile-manipulator”, i.e., using the computational model of the mobile-manipulator

$${}^v g_{kX} = g(\mathbf{x}_r(t), {}^\Omega \Theta_{kX}), \quad (14)$$

where ${}^v g_{kX}$ denotes the value of objective function at time-step k . Note that this equation represents two values for $X \in \{L, R\}$. The left superscript symbol v denotes the output value from a “virtual mobile-manipulator”.

Next, we make use of the following rule to update the location of beetle, inside the joint-space, toward a direction in which the value of the objective function is decreasing

$${}^\Omega \Theta'_{k+1} = \mathcal{P}_\Omega(\Theta_k - \delta_k(\lambda_k) \text{sign}({}^v g_{kL} - {}^v g_{kR}) \vec{\mathbf{b}}), \quad (15)$$

where ${}^\Omega \Theta'_{k+1}$ is the new location of beetle. $\delta_k(\lambda_k)$, is the actual step-size, also a hyperparameter of our algorithm. The dependancy of step-size on antenna-length is described later. Next, we evaluate the objective function at the new

location ${}^{\Omega}\Theta'_{k+1}$, again using “virtual mobile-manipulator”

$${}^v g'_{k+1} = g(\mathbf{x}_r(t), {}^{\Omega}\Theta'_{k+1}). \quad (16)$$

If the value of objective function at new location has improved (i.e., the new value g'_{k+1} is smaller than g_k), then the mobile-manipulator changes its states to ${}^{\Omega}\Theta'_{k+1}$, otherwise it retain its current state

$$\Theta_{k+1} = \begin{cases} {}^{\Omega}\Theta'_{k+1} & \text{if } {}^v g'_{k+1} < g_k \\ \Theta_k & \text{if } {}^v g'_{k+1} \geq g_k. \end{cases} \quad (17)$$

Similarly, we assign the new value to g_{k+1} for use during next iteration

$$g_{k+1} = \begin{cases} {}^v g'_{k+1} & \text{if } {}^v g'_{k+1} < g_k \\ g_k & \text{if } {}^v g'_{k+1} \geq g_k. \end{cases} \quad (18)$$

After updating the state of mobile-manipulator to Θ_{k+1} , the iterative procedure is repeated for the next time-steps. The steps of the algorithm are systematically listed in Algorithm 1.

The value of hyper-parameters λ_k and $\delta_k(\lambda_k)$, affects the speed of convergence. Care must be taken to choose the initial value of these parameters. By analyzing the simulation results, we found that the following rules help regulate the values of antenna-length and step-size

$$\begin{aligned} \lambda_k &= \sqrt{g'_k} \\ \delta_k(\lambda_k) &= \lambda_k. \end{aligned}$$

The above relations control the antennae length such that the step-size is directly proportional to the value of the objective function. Such behavior is necessary to prevent the overshooting of the search particle. For c_1 and c_2 , we propose the following rules for fast convergence.

$$\begin{aligned} c_1 &\propto T_s \\ c_2 &\in [1, 3] \end{aligned}$$

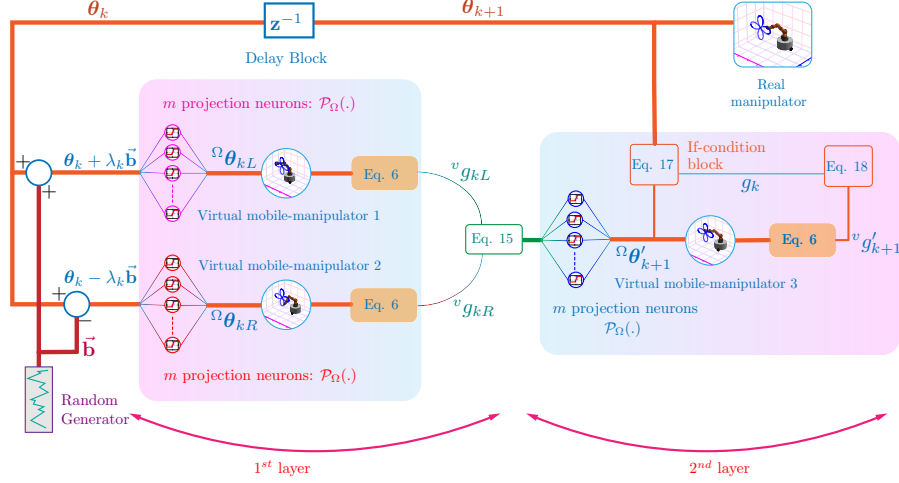


Figure 2: RNN architecture of the BAORNN algorithm. The architecture implement the working of algorithm formulated in Section 3.1.

where T_s is the sampling time of the control loop.

Fig. 2 shows the topology of RNN corresponding to the BAORNN algo-rithm. The network topology is called an RNN because it has delay blocks, which act as a memory of the network and use the values from past iterations to calculate the values in the next iterations. There are a total of 2 layers. The first layer represents the equations (12), (13), and (14). The seconds layer represent the equations (15), (17), and (18). There is a temporal-feedback con-nection between the second layer and the first layer through a delay block. The RNN have a total of $3m + 6$ neurons, out of which $3m$ are projection neurons, shown as small circles, corresponding to the projection function $\mathcal{P}_\Omega(.)$. 3 neurons, shown as circles with manipulator images, implement the functionality of “virtual robots”. Additionally, 3 more neurons, shown as rectangular curved boxes (in orange), evaluate the objective function $g(.)$. The diagram also contains an “If-condition” blocks to implement the functionality of (17) and (18). The proposed algorithm require very small computation power since most of the steps can be executed by using very elementary floating-point operations. Espically, for modern embedded processors since they have dedicated hardware

Algorithm 1: BAORNN algorithm - Tracking control of mobile-manipulator

Input: Kinematic model $f(\cdot)$ of the mobile-manipulator, a reference trajectory

$\mathbf{x}_r(t) \in \mathbb{R}^n$, an objective function $g(\mathbf{x}_r, \boldsymbol{\Theta})$.

Output: An optimal joint-space trajectory $\boldsymbol{\Theta}_r(t)$.

$\boldsymbol{\Theta}_0 \leftarrow$ Initial joint coordinates

$k \leftarrow 0$

k_{stop} maximum allowed time-steps

while $k < k_{stop}$ **do**

 Generate a uniformly distributed normalized random direction vectors,

$\vec{\mathbf{b}} \in \mathbb{R}^{m+3}$ in the joint-space.

 Use the direction vector to calculate the antennae location using (12).

 Use the projection function $\mathcal{P}_\Omega(\cdot)$ to restrict the location of each antennae on the constrained set Ω as defined in (13).

 Evaluate the objective function at both location using "Virtual mobile-manipulators" as done in (14).

 Calculate the location of next step using the update rule in (15).

 Use (17) and (18) to update the value of current location and objective function.

$k \leftarrow k + 1$

end

unit for floating-point calculations.

Note that the difference between the virtual manipulator blocks and the real manipulator. The virtual manipulator is only able to test the effect of control actions in virtual space and therefore does not directly influence the movement of the real-manipulator. However, they have an indirectly control the movement of the real manipulator by feeding their output value to other blocks of the RNN.

3.2. Computational Cost

To prove the efficiency of BAORNN algorithm, we will analyze its computational complexity and prove that the number of mathematical operations required is a linear function of the number of joints, m , of manipulator. The first step in Algorithm 1 requires generation of $m + 3$ random numbers, which require $\alpha_1(m + 3)$ floating-point operations, where α operations are required to generate single random number. The second step of the Algorithm 1 needs $2(m + 3)$ additions and $m + 3$ multiplication operations, making a total $3(m + 3)$

floating-point operations. The third-step of the algorithm requires the projection of two vectors on constrained set Ω requiring $2(m + 3)$ comparisons (2 equality and $2(m+2)$ inequality comparisons) in total. The fourth step of the algorithm requires evaluation of objective function as given in (6) twice. The objective function is just euclidean distance of tracking error and it can be seen that its single evaluation requires about $\alpha_2 m + 3$ floating-point operations, where α_2 is a proportionality constant. Therefore, $2\alpha_2 m + 6$ floating-points operation are required for step 4 in total. The fifth step of the algorithm is the calculation of updated location and requires a total of about $4(m + 3)$ operations. The sixth step again requires the calculation of the objective function, i.e., $8m$ additional floating-point operations. The last step of the proposed algorithm requires the re-evaluation of objective function and additionally 2 comparison making a total of $\alpha_2 + 5$ floating-point operations. Combining all these makes the final operation counts to be $\alpha_1(m+3)+3(m+3)+2\alpha_2 m+6+4(m+3)+\alpha_2+5 = (\alpha_1 + 2\alpha_2 + 7)m + 3\alpha_1 + \alpha_2 + 32$. Thus proven that the complexity of the proposed algorithm is $O(m)$, i.e., linear function of the number of joints.

3.3. Theoretical Analysis

Lemma 1. *For the tracking control of a redundant mobile-manipulator, starting from an initial state Θ_0 the joint-space trajectory $\Theta_r(t)$ generated by BAORNN algorithm is stable, i.e.,*

$$g_{k+1} \leq g_k, \quad \forall \quad k \geq 0, \quad (19)$$

the values of objective function are decreasing.

Proof. See Lemma 1 of [58]. □

Lemma 2. *For the tracking control of a redundant mobile-manipulator, starting from an initial state Θ_0 the end-effector trajectory $F(\Theta_r(t))$ is convergent to the reference trajectory $\mathbf{x}_r(t)$, i.e.,*

$$f(\theta(t)) \rightarrow \mathbf{x}_r(t), \quad \text{as } t \rightarrow \infty. \quad (20)$$

Proof. See Theorem 1 of [58]. □

4. Simulation Results & Discussion

In this section, we will first describe the simulation platform; then, we will present the methodology of conducting the numerical analysis of the proposed algorithm followed by the discussion on the simulations results.

4.1. Simulation Studies

MATLAB Robotic System Toolbox [73] provides a simulation workbench for the IIWA-14 industrial robotic manipulator along with the 3D visualization. The simulated model is excellent in testing the performance of the control algorithms before actually implementing on real hardware. To systematically study the performance, we used three different reference trajectories: circular, rectangular, and rhodonea trajectories. These trajectories represent a wide range of task mostly required to be performed by an industrial mobile-manipulator. The structure of the robot is shown in Figure 1.

The equations describing the reference trajectories are as follow. The circular trajectory is generated using

$$\mathbf{x}_r^{\text{circle}}(t) = \vec{\mathbf{C}} + r \cos(2\pi t/T) \vec{\mathbf{A}} + r \sin(2\pi t/T) \vec{\mathbf{B}}.$$

where $\vec{\mathbf{C}}$ is the position vector to the center of the circle, $\vec{\mathbf{A}}$ and $\vec{\mathbf{B}}$ are two orthogonal unit vectors, i.e., $|\vec{\mathbf{A}}|_2 = 1$, $|\vec{\mathbf{B}}|_2 = 1$ and $\vec{\mathbf{A}} \perp \vec{\mathbf{B}}$. These vectors define the plane of the circle in 3D-space. r is the radius of the circle and T denotes the total time-duration for the circular trajectory. The rectangular trajectory is generated using

$$\mathbf{x}_r^{\text{rect}}(t) = \begin{cases} \frac{t_0-t}{t_0} \vec{\mathbf{a}} + \frac{t}{t_0} \vec{\mathbf{b}} & \text{when } 0 \leq t \leq t_0 \\ \frac{t_1-t}{t_1-t_0} \vec{\mathbf{b}} + \frac{t-t_0}{t_1-t_0} \vec{\mathbf{c}} & \text{when } t_0 < t \leq t_1 \\ \frac{t_2-t}{t_2-t_1} \vec{\mathbf{c}} + \frac{t-t_1}{t_2-t_1} \vec{\mathbf{d}} & \text{when } t_1 < t \leq t_2 \\ \frac{t_3-t}{t_3-t_2} \vec{\mathbf{d}} + \frac{t-t_2}{t_3-t_2} \vec{\mathbf{a}} & \text{when } t_2 < t \leq t_3. \end{cases}$$

where $\vec{\mathbf{a}}$, $\vec{\mathbf{b}}$, $\vec{\mathbf{c}}$, and $\vec{\mathbf{d}}$ are the corners of the rectangle and t_0 , t_1 , t_2 , and t_3 are the time to reach the corners of the rectangle trajectory. For generating the rhodonea trajectory we used following equation

$$\mathbf{x}_r^{\text{rhod}}(t) = \vec{\mathbf{D}} + \begin{bmatrix} \sigma \cos(2t) \cos(t) \\ 0 \\ \sigma \cos(2t) \sin(t) \end{bmatrix}$$

where $\vec{\mathbf{D}}$ is the center of the rhodonea trajectory and σ control the size of trajectory.

4.2. Trajectory Tracking Results

Figure 3 shows the results for the circular trajectory using the BAORNN algorithm. It can be seen that the algorithm is effective in deriving the end-effector along the reference trajectory while moving the manipulator's links and the mobile platform simultaneously. Figure 3(a) shows the trajectory of the links of the manipulator at all time-instants, starting from the home configuration of the IIWA14 robot. For the ease of visualization, only shadows of the mobile platform are shown on the ground, at a few time-instants. The reference trajectory is shown as a blue line. The profile of the joint-angles is also shown in Figure 3(b). Similarly, the profile of the platform's states, as defined in Figure 1(b), are shown in Figure 3(c). The cartesian task-space coordinates of the end-effector are shown in Figure 3(d). The profile of the tracking error is shown in Figure 3(e). The tracking error is defined as

$$\mathbf{e}(t) = F(\Theta_r(t)) - \mathbf{x}_r(t), \quad (21)$$

where $\mathbf{e}(t) = [e_x(t) \ e_y(t) \ e_z(t)]$. It can be seen that the tracking error is quite large at the beginning; this is because the manipulator starts from the home configuration and it takes some time for the algorithm to make the end-effector reach near the reference trajectory. Once the end-effector reaches the reference trajectory, it closely follows it, and the tracking error is reduced to zero. Along with the tracking error, Figure 3(f) also shows the value of the objective function

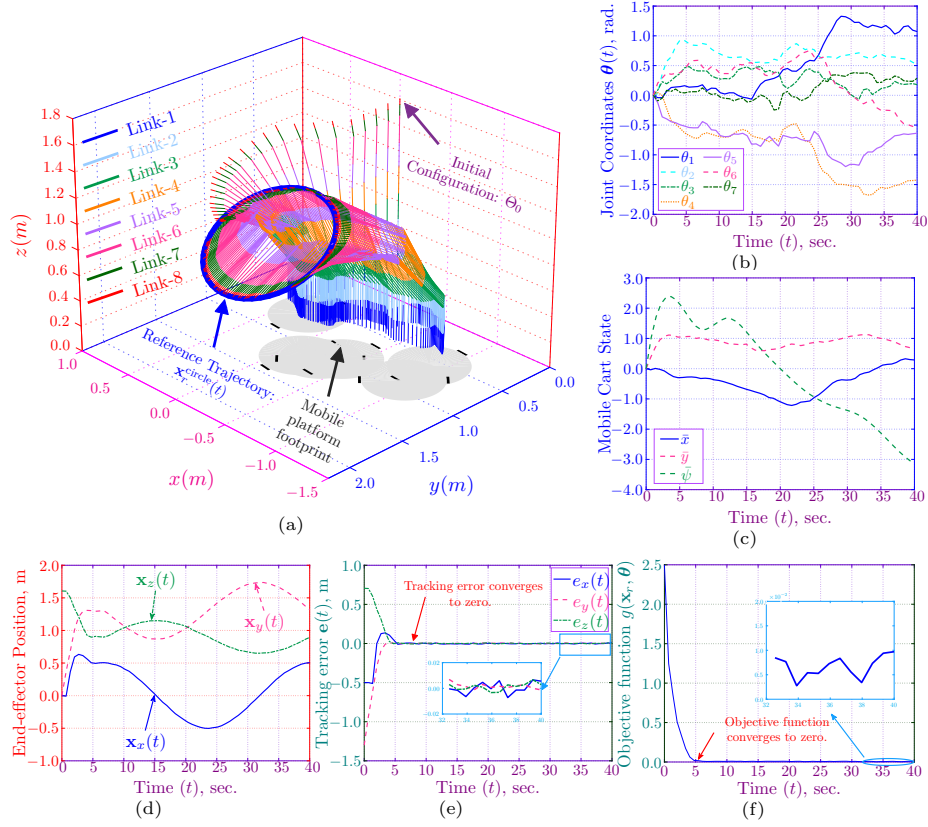


Figure 3: Simulation results for circular trajectory tracking. (a) 3D motion of the manipulator platform and manipulator's link. The footprint of the mobile platform on the ground are also shown. (b) Profile of joint-angles of the manipulator. (c) Profile of the states of mobile platform. (d) Profile of cartesian task-space Coordinates of the end-effector. (e) Profile of tracking error as defined in (21). (f) Profile of objective function as defined in (11).

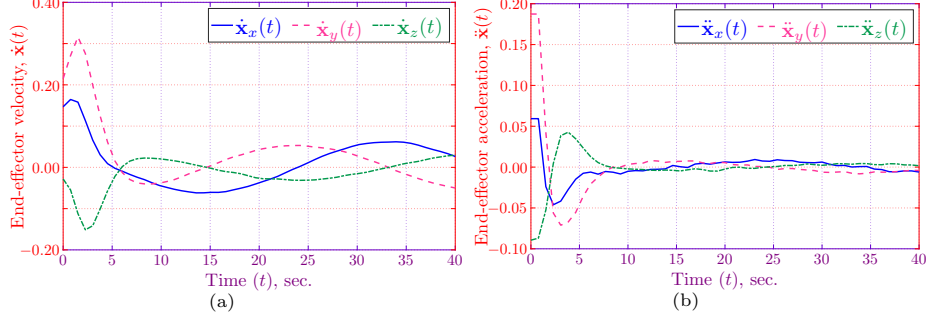


Figure 4: The dynamic response for the circular trajectory tracking experiments. (a) Shows the velocity response, (b) shows the acceleration response. The responses show that the velocity and acceleration remain within a small range and does not produce excessive stress on the system.

as defined in (11). It also shows that the tracking error converges to zero very quickly. These results demonstrate that the proposed algorithm is very effective in tracking the rectangular trajectory. It must be noted that the tracking error and objective function shows a ripply profile. Such behavior is characteristic of metaheuristic algorithms because of their stochastic nature, i.e., using a random direction vector \vec{b} in each iteration. The dynamic response of the end-effector trajectory is also shown in Figure 4. Figure 4(a) shows the profile of velocity and Figure 4(b) shows the profile of acceleration. It can be seen that the value of velocity and acceleration remain very small during the tracking of reference trajectory. Since the forces and torques created by end-effectors are directly proportional to the acceleration, therefore, the proposed algorithm is effectively able to limit the amount of jointing torques and able to operate the manipulator without the risk of damaging.

Similar results for circular and rhodonea trajectories are shown in Figure 5 and Figure 7 respectively. It can be seen that similar to the results of circular trajectory, the BAORNN algorithms is effectively able to track the reference trajectories. The position of the end-effector closely matches the reference trajectories (shown in blue) in Figure 5(a) and Figure 7(a). The tracking error

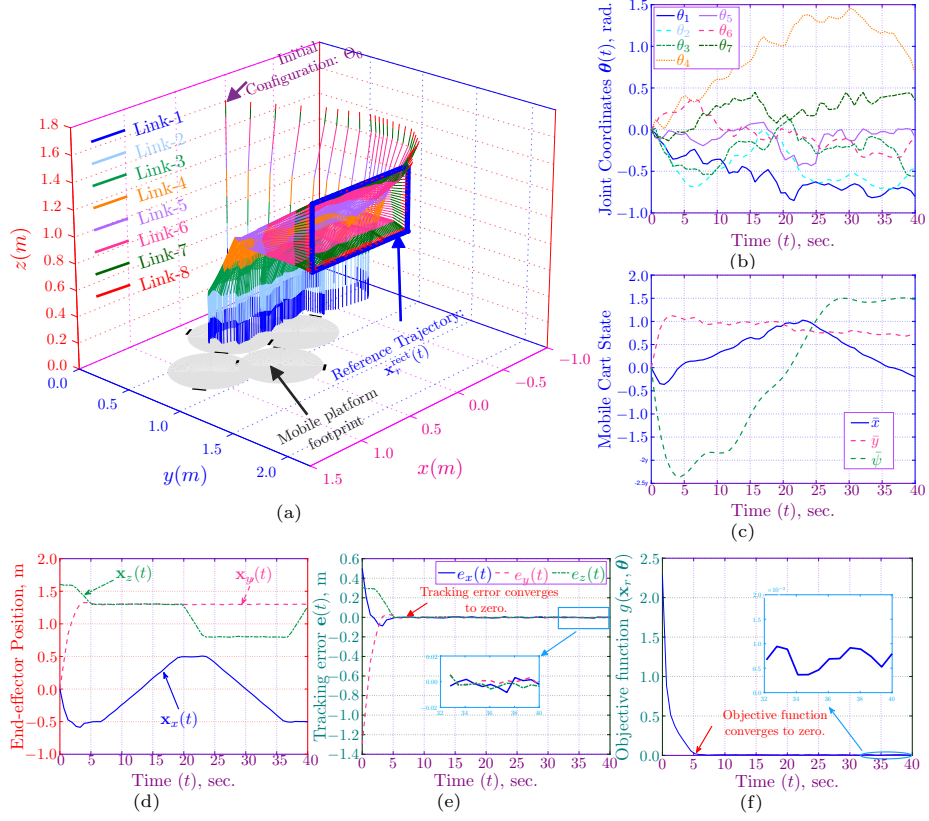


Figure 5: Simulation results for circular trajectory tracking. (a) 3D motion of the manipulator platform and manipulator's link. The footprint of the mobile platform on the ground are also shown. (b) Profile of joint-angles of the manipulator. (c) Profile of the states of mobile platform. (d) Profile of cartesian task-space Coordinates of the end-effector. (e) Profile of tracking error as defined in (21). (f) Profile of objective function as defined in (11).

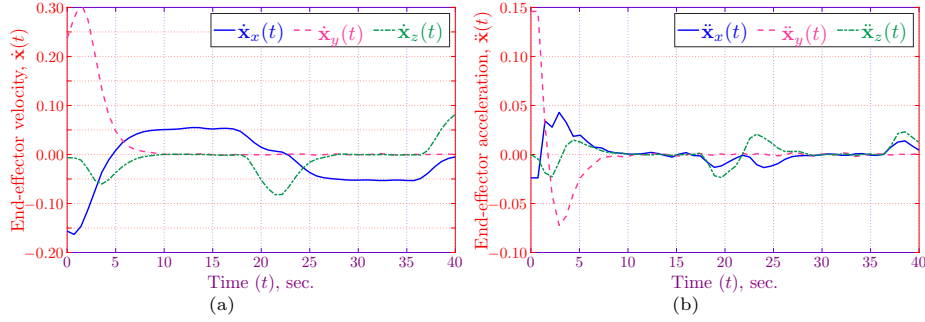


Figure 6: The dynamic response for the rectangular trajectory tracking experiments. (a) Shows the velocity response, (b) shows the acceleration response. The responses show that the velocity and acceleration remain within a small range and does not produce excessive stress on the system.

and objective function values are also shown in Figure 5(e,f) and Figure 7(e,f) respectively. These plot also shows that the tracking error initially start from a very large value and then quickly converges to zero. Figure 6 and Figure 8 shows the dynamic response for the rectangular and rhodonea trajectories respectively. It can be seen that, similar to the case of circular trajectory, the velocity and acceleration remain bounded within small range, thereby increasing the safety of the control algorithm. To visualize the performance, Figure 9 shows the 3D model of the mobile-manipulator while following the reference trajectories.

Similarly, we performed the same set of simulations using PSO algorithm to compare the performance and prove the numerical efficiency of the pro-posed algorithm. The results obtained using PSO algorithm are shown in Figure 10. However, the tracking error is a bit higher as compared to the proposed BAORNN algorithm. Figure 10(a)-(c) show the tracking performance of PSO for the circular reference trajectory. Figure 10(a) shows the profile of end-effector's coordinates. It can be seen that, the profile of PSO algorithm shows noisy behavior as compared to the profile of BAORNN algorithm as shown in Figure 3(d). Figure 10(b) and Figure 10(c) shows the profile of objective function and tracking error respectively. It can be seen that the magnitude of objec-

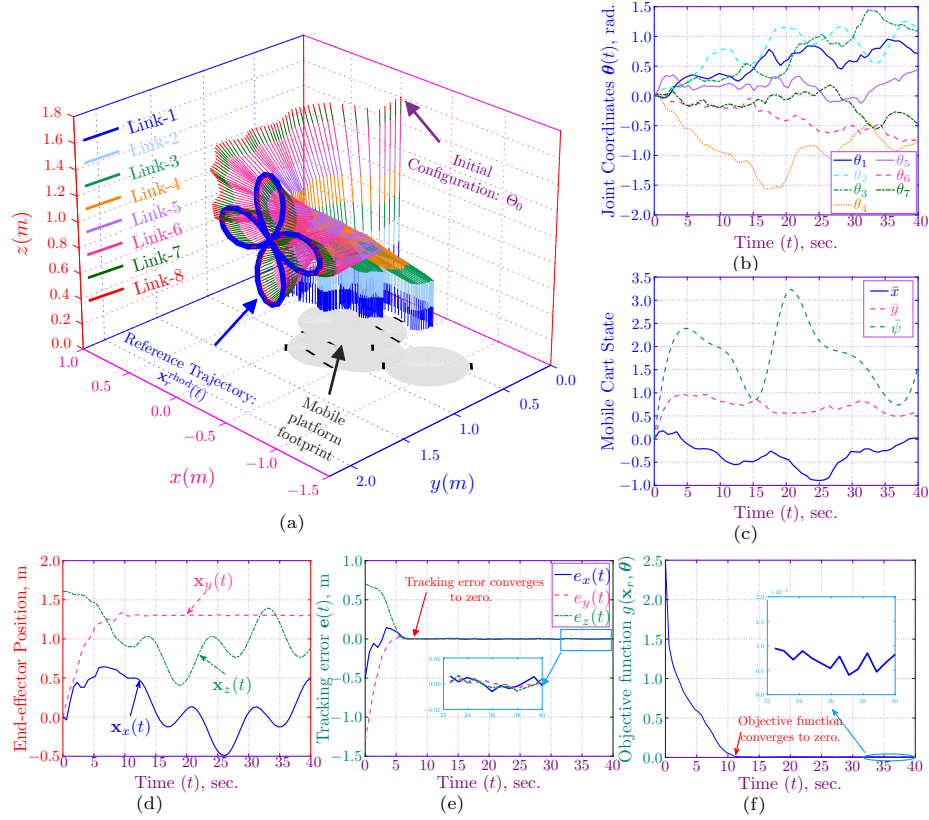


Figure 7: Simulation results for rhodonea trajectory tracking. (a) 3D motion of the manipulator platform and manipulator's link. The footprint of the mobile platform on the ground are also shown. (b) Profile of joint-angles of the manipulator. (c) Profile of the states of mobile platform. (d) Profile of cartesian task-space Coordinates of the end-effector. (e) Profile of tracking error as defined in (21). (f) Profile of objective function as defined in (11).

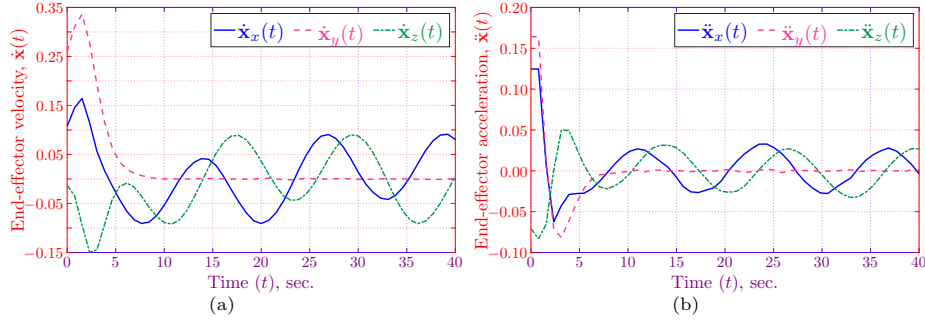


Figure 8: The dynamic response for the rhodonea trajectory tracking experiments. (a) Shows the velocity response, (b) shows the acceleration response. The responses show that the velocity and acceleration remain within a small range and does not produce excessive stress on the system.

tive function and tracking error is also higher as compared to the BAORNN algorithm as shown in Figure 3(e)-(f). A similar trend can be observed for the circular and rhodonea trajectory in Figure 10(d)-(f) and Figure 10(g)-(i) respectively. To prove the computation efficiency of the proposed algorithm, we compared the execution times of both algorithm for calculating the complete trajectory. The comparison results are summarized in Table 1. It can be seen that the proposed algorithm is able to execute extremely fast and able to plan the trajectory within few seconds, whereas, PSO takes a very long time making it unsuitable for real-time control of mobile manipulator.

5. Conclusion

In this paper, we proposed a metaheuristic algorithm to address the problem of tracking control for a redundant mobile-manipulator. We formulated the tracking control as a constrained optimization problem. The formulated optimization problem also account for the joint-angle limits, motor-speed limits, and nonholonomic constraints on the mobile platform. To numerically solve the optimization problem, we propose a RNN based on metaheuristic algorithm. A key feature of the proposed algorithm is the introduction of the concept of “virtual



Figure 9: 3D visualization of the simulated models. (a) Images for circular reference trajectory, (b) images for rectangular reference trajectory and (c) images for rhodonea reference trajectory.

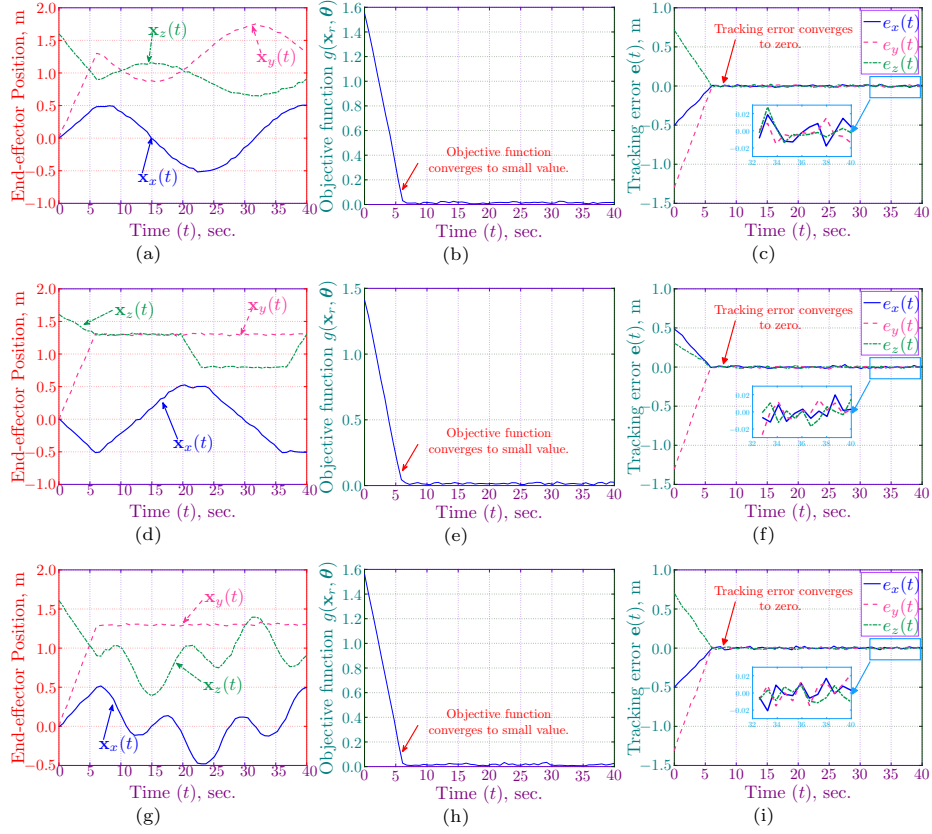


Figure 10: Results of PSO optimization algorithm for the tracking control problem. (a) the profile of end-effector coordinates in task-space, (b) value of objective function, (c) profile of tracking errors for the circular trajectory. (d)-(f) show the similar results for rectangular trajectory and (g)-(i) show the results for rhodonea path.

Table 1: Comparasion of BAORNN and PSO algorithm.

Trajectory	Circular		Rectangular		Rhodonea	
Algorithm	BAORNN	PSO	BAORNN	PSO	BAORNN	PSO
Avg. $g(.)^*$	0.0074	0.0164	0.0074	0.0176	0.0072	0.0164
Avg. $e_x(.)^{**}$	0.0040	0.0081	0.0029	0.0078	0.0033	0.0079
Avg. $e_y(.)^{**}$	0.0025	0.0059	0.0027	0.0081	0.0024	0.0085
Avg. $e_z(.)^{**}$	0.0028	0.0075	0.0028	0.0093	0.0027	0.0083
Exec. time [†]	4.9	375	4.05	488	3.26	468

* Average value of objective function $g(.)$ as defined in (6).

** Average value of tracking error along an axis as defined in (21).

[†] Average execution time in seconds as measured in MATLAB.

robots”; The virtual mobile-manipulators help in anticipating the task-space effect of any joint-space action without moving the real mobile-manipulator. We presented the theoretical treatment of the proposed algorithm. The simulation results using IIWA14, a 7-DOF manipulator, are also performed using different reference trajectories. The simulation results demonstrate the efficacy of the proposed algorithm in tracking an arbitrarily shaped reference trajectory.

References

- [1] W. Ye, Z. Li, C. Yang, J. Sun, C.-Y. Su, R. Lu, Vision-based human tracking control of a wheeled inverted pendulum robot, IEEE transactions on cybernetics 46 (11) (2015) 2423–2434.
- [2] D. Zhou, M. Shi, F. Chao, C.-M. Lin, L. Yang, C. Shang, C. Zhou, Use of human gestures for controlling a mobile robot via adaptive cmac network and fuzzy logic controller, Neurocomputing 282 (2018) 218–231.
- [3] W. Ye, Z. Li, C. Yang, J. Sun, C.-Y. Su, R. Lu, Vision-based human tracking control of a wheeled inverted pendulum robot, IEEE transactions on cybernetics 46 (11) (2015) 2423–2434.

- [4] F. Baghernezhad, K. Khorasani, Computationally intelligent strategies for robust fault detection, isolation, and identification of mobile robots, *Neurocomputing* 171 (2016) 335–346.
- [5] C. Yang, Y. Jiang, Z. Li, W. He, C.-Y. Su, Neural control of bimanual robots with guaranteed global stability and motion precision, *IEEE Transactions on Industrial Informatics* 13 (3) (2016) 1162–1171.
- [6] Z. Xu, S. Li, X. Zhou, T. Cheng, Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties, *Neurocomputing* 357 (2019) 271–281.
- [7] F. Luan, J. Na, Y. Huang, G. Gao, Adaptive neural network control for robotic manipulators with guaranteed finite-time convergence, *Neurocomputing* 337 (2019) 153–164.
- [8] D. Chen, Y. Zhang, S. Li, Tracking control of robot manipulators with unknown models: A jacobian-matrix-adaption method, *IEEE Transactions on Industrial Informatics* 14 (7) (2017) 3044–3053.
- [9] L. Jin, S. Li, X. Luo, Y. Li, B. Qin, Neural dynamics for cooperative control of redundant robot manipulators, *IEEE Transactions on Industrial Informatics* 14 (9) (2018) 3812–3821.
- [10] A. M. Zanchettin, L. Bascetta, P. Rocco, Achieving humanlike motion: Resolving redundancy for anthropomorphic industrial manipulators, *IEEE Robotics & Automation Magazine* 20 (4) (2013) 131–138.
- [11] D. Guo, Y. Zhang, Acceleration-level inequality-based man scheme for obstacle avoidance of redundant robot manipulators, *IEEE Transactions on Industrial Electronics* 61 (12) (2014) 6903–6914.
- [12] H. M. La, R. Lim, W. Sheng, Multirobot cooperative learning for predator avoidance, *IEEE Transactions on Control Systems Technology* 23 (1) (2014) 52–63.

- [13] H. Ding, S. K. Tso, A fully neural-network-based planning scheme for torque minimization of redundant manipulators, *IEEE Transactions on Industrial Electronics* 46 (1) (1999) 199–206.
- [14] G. Tevatia, S. Schaal, Inverse kinematics for humanoid robots, in: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, Vol. 1, IEEE, 2000, pp. 294–299.
- [15] A. Goldenberg, B. Benhabib, R. Fenton, A complete generalized solution to the inverse kinematics of robots, *IEEE Journal on Robotics and Automation* 1 (1) (1985) 14–20.
- [16] S. Li, S. Chen, B. Liu, Y. Li, Y. Liang, Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks, *Neurocomputing* 91 (2012) 1–10.
- [17] L. Jin, S. Li, H. M. La, X. Luo, Manipulability optimization of redundant manipulators using dynamic neural networks, *IEEE Transactions on Industrial Electronics* 64 (6) (2017) 4710–4720.
- [18] C. Yang, H. Wu, Z. Li, W. He, N. Wang, C.-Y. Su, Mind control of a robotic arm with visual fusion technology, *IEEE Transactions on Industrial Informatics* 14 (9) (2017) 3822–3830.
- [19] W. He, H. Huang, S. S. Ge, Adaptive neural network control of a robotic manipulator with time-varying output constraints, *IEEE transactions on cybernetics* 47 (10) (2017) 3136–3147.
- [20] H.-q. Wang, B. Chen, C. Lin, Adaptive neural tracking control for a class of stochastic nonlinear systems, *International Journal of Robust and Nonlinear Control* 24 (7) (2014) 1262–1280.
- [21] H. Wang, X. Liu, K. Liu, Robust adaptive neural tracking control for a class of stochastic nonlinear interconnected systems, *IEEE transactions on neural networks and learning systems* 27 (3) (2015) 510–523.

- [22] B. Liao, W. Liu, Pseudoinverse-type bi-criteria minimization scheme for redundancy resolution of robot manipulators, *Robotica* 33 (10) (2015) 2100–2113.
- [23] L. Jin, Y. Zhang, Discrete-time zhang neural network of o (τ 3) pattern for time-varying matrix pseudoinversion with application to manipulator motion generation, *Neurocomputing* 142 (2014) 165–173.
- [24] A. Liegeois, Automatic supervisory control of the configuration and behaviour of multibody mechanisms, *IEEE Transactions on systems, man and cybernetics* 7 (12) (1977) 868–871.
- [25] C. A. Klein, C.-H. Huang, Review of pseudoinverse control for use with kinematically redundant manipulators, *IEEE Transactions on Systems, Man, and Cybernetics* (2) (1983) 245–250.
- [26] D. Chen, S. Li, Q. Wu, X. Luo, New disturbance rejection constraint for redundant robot manipulators: An optimization perspective, *IEEE Transactions on Industrial Informatics* (2019).
- [27] S.-H. Cha, T. Lasky, S. Velinsky, Kinematic redundancy resolution for serial-parallel manipulators via local optimization including joint constraints, *Mechanics based design of structures and machines* 34 (2) (2006) 213–239.
- [28] J. Woolfrey, W. Lu, D. Liu, A control method for joint torque minimization of redundant manipulators handling large external forces, *Journal of Intelligent & Robotic Systems* (2019) 1–14.
- [29] W. He, Z. Yan, Y. Sun, Y. Ou, C. Sun, Neural-learning-based control for a constrained robotic manipulator with flexible joints, *IEEE transactions on neural networks and learning systems* 29 (12) (2018) 5993–6003.
- [30] H. Wang, S. Kang, Adaptive neural command filtered tracking control for flexible robotic manipulator with input dead-zone, *IEEE Access* 7 (2019) 22675–22683.

- [31] B. Luo, D. Liu, T. Huang, D. Wang, Model-free optimal tracking control via critic-only q-learning, *IEEE transactions on neural networks and learning systems* 27 (10) (2016) 2134–2144.
- [32] T. R. Wanasinghe, G. K. Mann, R. G. Gosine, A jacobian free approach for multi-robot relative localization, in: *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, IEEE, 2014, pp. 1–6.
- [33] J. Na, X. Ren, D. Zheng, Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer, *IEEE transactions on neural networks and learning systems* 24 (3) (2013) 370–382.
- [34] W. He, Z. Yin, C. Sun, Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier lyapunov function, *IEEE Transactions on Cybernetics* 47 (7) (2016) 1641–1651.
- [35] C. Yang, Y. Jiang, J. Na, Z. Li, L. Cheng, C.-Y. Su, Finite-time convergence adaptive fuzzy control for dual-arm robot with unknown kinematics and dynamics, *IEEE Transactions on Fuzzy Systems* 27 (3) (2018) 574–588.
- [36] J. Na, B. Jing, Y. Huang, G. Gao, C. Zhang, Unknown system dynamics estimator for motion control of nonlinear robotic systems, *IEEE Transactions on Industrial Electronics* (2019).
- [37] H. Wang, Y. Zou, P. X. Liu, X. Liu, Robust fuzzy adaptive funnel control of nonlinear systems with dynamic uncertainties, *Neurocomputing* 314 (2018) 299–309.
- [38] C. Yang, Y. Jiang, W. He, J. Na, Z. Li, B. Xu, Adaptive parameter estimation and control design for robot manipulators with finite-time convergence, *IEEE Transactions on Industrial Electronics* 65 (10) (2018) 8112–8123.
- [39] J. Na, M. N. Mahyuddin, G. Herrmann, X. Ren, P. Barber, Robust adaptive finite-time parameter estimation and control for robotic systems, *International Journal of Robust and Nonlinear Control* 25 (16) (2015) 3045–3071.

- [40] D. Connell, H. M. La, Dynamic path planning and replanning for mobile robots using rrt, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2017, pp. 1429–1434.
- [41] H. Xiao, Z. Li, C. Yang, L. Zhang, P. Yuan, L. Ding, T. Wang, Robust stabilization of a wheeled mobile robot using model predictive control based on neurodynamics optimization, *IEEE Transactions on Industrial Electronics* 64 (1) (2016) 505–516.
- [42] D. Connell, H. Manh La, Extended rapidly exploring random tree-based dynamic path planning and replanning for mobile robots, *International Journal of Advanced Robotic Systems* 15 (3) (2018) 1729881418773874.
- [43] M. S. Miah, W. Gueaieb, Rfid-based mobile robot trajectory tracking and point stabilization through on-line neighboring optimal control, *Journal of Intelligent & Robotic Systems* 78 (3-4) (2015) 377–399.
- [44] D. Chwa, Fuzzy adaptive tracking control of wheeled mobile robots with state-dependent kinematic and dynamic disturbances, *IEEE transactions on Fuzzy Systems* 20 (3) (2011) 587–593.
- [45] B. Deepak, D. R. Parhi, S. Kundu, Innate immune based path planner of an autonomous mobile robot, *Procedia Engineering* 38 (2012) 2663–2671.
- [46] D. Chen, Y. Zhang, Robust zeroing neural-dynamics and its time-varying disturbances suppression model applied to mobile robot manipulators, *IEEE transactions on neural networks and learning systems* 29 (9) (2017) 4385–4397.
- [47] J. A. Parejo, A. Ruiz-Cortés, S. Lozano, P. Fernandez, Metaheuristic optimization frameworks: a survey and benchmarking, *Soft Computing* 16 (3) (2012) 527–561.
- [48] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*, John Wiley & Sons, 2010.

- [49] Z. Ren, P. Li, J. Fang, H. Li, Q. Chen, Sba: an efficient algorithm for address assignment in zigbee networks, *Wireless personal communications* 71 (1) (2013) 719–734.
- [50] J. Fang, L. Zhang, H. Li, Two-dimensional pattern-coupled sparse bayesian learning via generalized approximate message passing, *IEEE Transactions on Image Processing* 25 (6) (2016) 2920–2930.
- [51] J. Fang, H. Li, Distributed estimation of gauss-markov random fields with one-bit quantized data, *IEEE Signal Processing Letters* 17 (5) (2010) 449–452.
- [52] J. Fang, Y. Shen, F. Li, H. Li, Z. Chen, Support knowledge-aided sparse bayesian learning for compressed sensing, in: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2015, pp. 3786–3790.
- [53] N. Razmjooy, M. Khalilpour, M. Ramezani, A new meta-heuristic optimization algorithm inspired by fifa world cup competitions: theory and its application in pid designing for avr system, *Journal of Control, Automation and Electrical Systems* 27 (4) (2016) 419–440.
- [54] A. Namadchian, M. Ramezani, N. Razmjooy, A new meta-heuristic algorithm for optimization based on variance reduction of gaussian distribution, *Majlesi Journal of Electrical Engineering* 10 (4) (2016) 49.
- [55] N. Razmjooy, M. Ramezani, An improved quantum evolutionary algorithm based on invasive weed optimization, *Indian J. Sci. Res* 4 (2) (2014) 413–422.
- [56] N. Razmjooy, M. Ramezani, Training wavelet neural networks using hybrid particle swarm optimization and gravitational search algorithm for system identification, *International Journal of Mechatronics, Electrical and Computer Technology* 6 (21) (2016) 2987–2997.

- [57] X. Jiang, S. Li, Bas: beetle antennae search algorithm for optimization problems, arXiv preprint arXiv:1710.10724 (2017).
- [58] Y. Zhang, S. Li, B. Xu, Convergence analysis of beetle antennae search algorithm and its applications, arXiv preprint arXiv:1904.02397 (2019).
- [59] Z. Zhu, Z. Zhang, W. Man, X. Tong, J. Qiu, F. Li, A new beetle antennae search algorithm for multi-objective energy management in microgrid, in: 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2018, pp. 1599–1603.
- [60] X. Yin, Y. Ma, Aggregation service function chain mapping plan based on beetle antennae search algorithm, in: Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering, ACM, 2018, pp. 225–230.
- [61] Y. Shi, et al., Particle swarm optimization: developments, applications and resources, in: Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546), Vol. 1, IEEE, 2001, pp. 81–86.
- [62] P. Moallem, N. Razmjooy, Optimal threshold computing in automatic image thresholding using adaptive particle swarm optimization, *Journal of applied research and technology* 10 (5) (2012) 703–712.
- [63] N. Razmjooy, M. Ramezani, Training wavelet neural networks using hybrid particle swarm optimization and gravitational search algorithm for system identification, *International Journal of Mechatronics, Electrical and Computer Technology* 6 (21) (2016) 2987–2997.
- [64] H. Zhang, H. Jin, Z. Liu, Y. Liu, Y. Zhu, J. Zhao, Real-time kinematic control for redundant manipulators in a time-varying environment: Multiple-dynamic obstacle avoidance and fast tracking of a moving object, *IEEE Trans. on Ind. Informatics* (2019).

- [65] Y. Zhang, S. Li, J. Zou, A. H. Khan, A passivity-based approach for kinematic control of redundant manipulators with constraints, *IEEE Trans. on Ind. Informatics* (2019).
- [66] P. Qi, C. Liu, A. Ataka, H.-K. Lam, K. Althoefer, Kinematic control of continuum manipulators using a fuzzy-model-based approach, *IEEE Trans. on Ind. Electron.* 63 (8) (2016) 5022–5035.
- [67] A. Menon, R. Prakash, L. Behera, Adaptive critic based optimal kinematic control for a robot manipulator, in: 2019 Intl. Conf. on Robot. and Autom. (ICRA), IEEE, 2019, pp. 1316–1322.
- [68] G. Wu, Kinematic analysis and optimal design of a wall-mounted four-limb parallel schönflies-motion robot for pick-and-place operations, *Journ. of Intelligent & Robotic Syst.* 85 (3-4) (2017) 663–677.
- [69] I. Al-Naimi, A. Taeim, N. Alajdah, Fully-automated parallel-kinematic robot for multitask ind. operations, in: 2018 15th Intl. Multi-Conf. on Syst., Signals & Devices, IEEE, 2018, pp. 390–395.
- [70] O. Hock, J. Sedo, Inverse kinematics using transposition method for robotic arm, in: 2018 ELEKTRO, IEEE, 2018, pp. 1–5.
- [71] Y. Zhang, X. Yan, D. Chen, D. Guo, W. Li, Qp-based refined manipulability-maximizing scheme for coordinated motion planning and control of physically constrained wheeled mobile redundant manipulators, *Nonlinear Dynamics* 85 (1) (2016) 245–261.
- [72] I. Anvari, Non-holonomic differential drive mobile robot control & design: Critical dynamics and coupling constraints, Ph.D. thesis, Arizona State University (2013).
- [73] P. I. Corke, et al., A robotics toolbox for matlab, *IEEE Robotics & Automation Magazine* 3 (1) (1996) 24–32.